Experiment Control System

&

Electronics Upgrade

Clara Gaspar
on behalf of the ECS team:
CERN, Marseille, etc.
October 2015

ECS & Upgrade Electronics

- **ECS Design doesn't change (in principle)**
  - Same tools:
    - Communications: DIM
    - Supervision: WinCC-OA
    - Sequencing and Automation (FSM): SMI++
  - Same philosophy:
    - Generic tools to describe the hardware (FwHw)
      - Board Types -> "chips" -> registers => Boards
    - Operation tools to Configure/Monitor boards
      - "Recipes" for different configuration modes
      - Stored in Configuration DB
  - New electronics will be interfaced like before

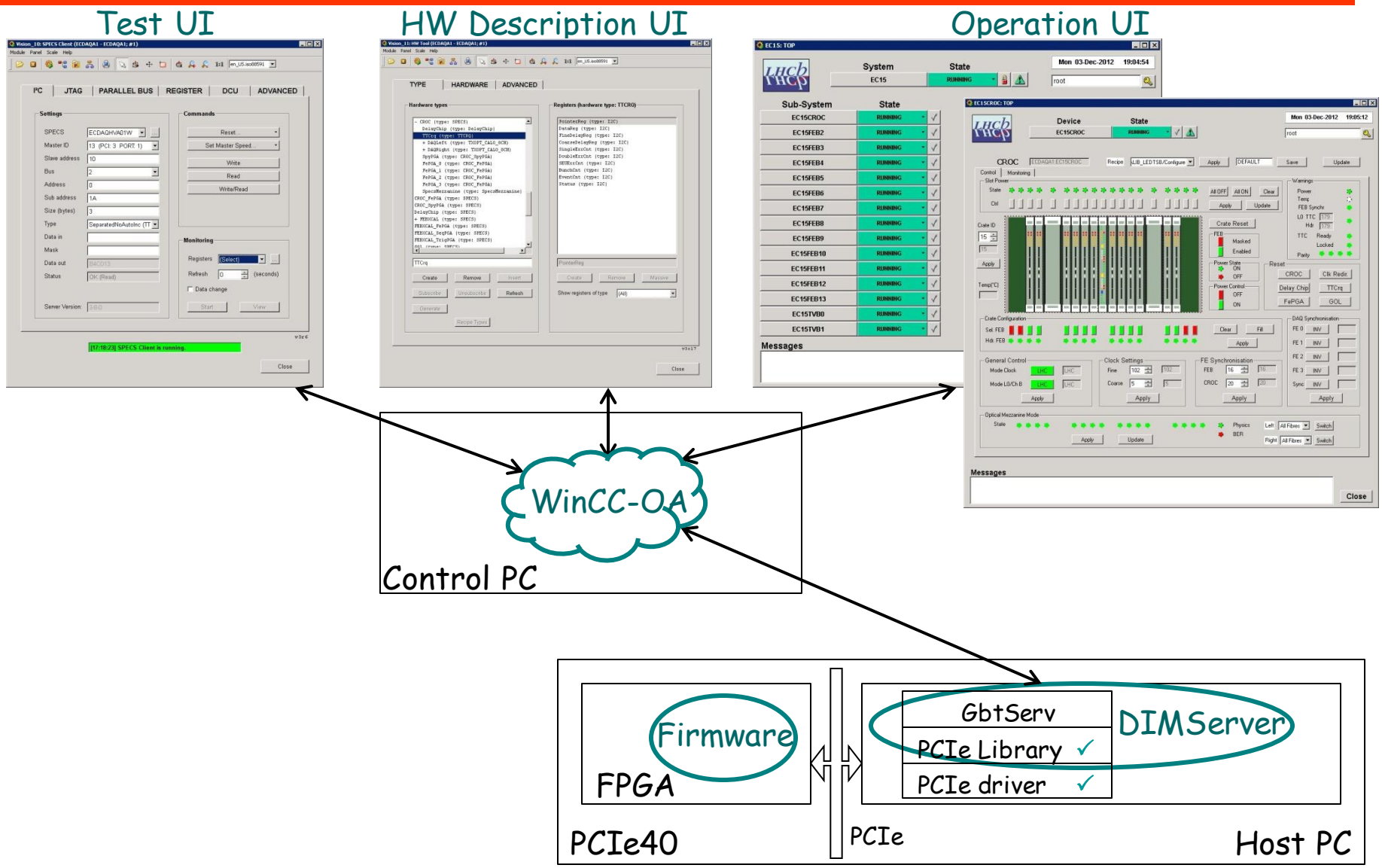
Electronics Interfaces

- **Two types of boards to control:**
  - Back-End boards (Readout Boards, etc.)
    - Physically only one board type (PCIe40):
    - But logically different types
      TELL40, S-ODIN, SOL40
    - (Past equivalent: fwCcpc)
  - Front-End Boards
    - Sub-detector Specific
    - Several protocols available (via GBT-SCA)
    - (Past equivalent: fwSpecs)

BE Board Interface

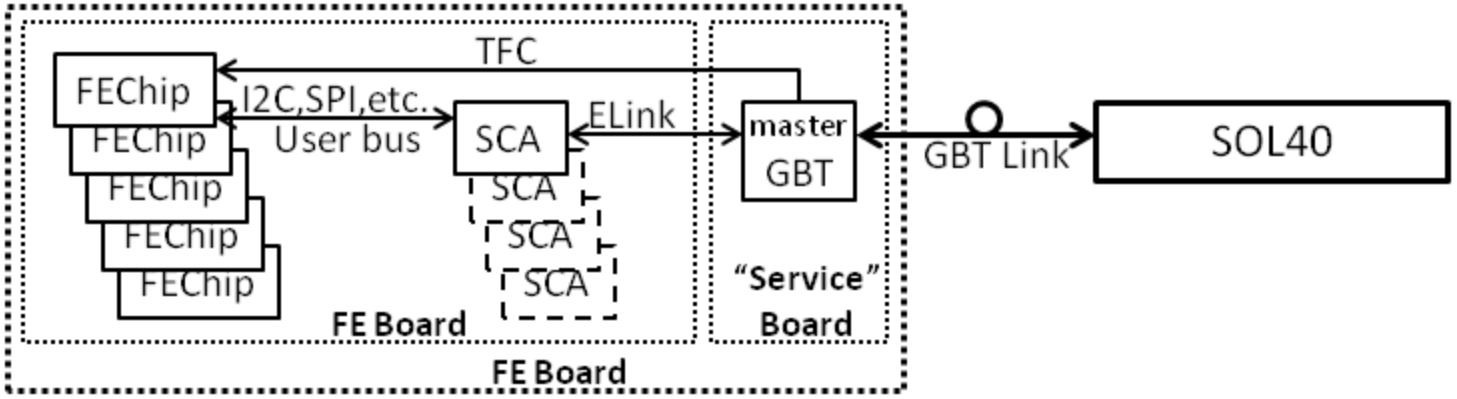
- **Physically only one board type (PCIe40) for: TELL40, S-ODIN, SOL40, etc.**
  - But logically different types (different firmware -> different “registers”)
- **Tools will be provided centrally:**
  - Low-level libraries and command-line tools:
    - Allow accessing the different registers (PCIexpress)
  - A DIM server:
    - Will implement higher-level commands to configure and monitor the board components
  - A WinCC-OA component
    - Providing the high-level description and access of all electronics components

ECS BE Dataflow



FE Board Interface

Interface to FE Electronics

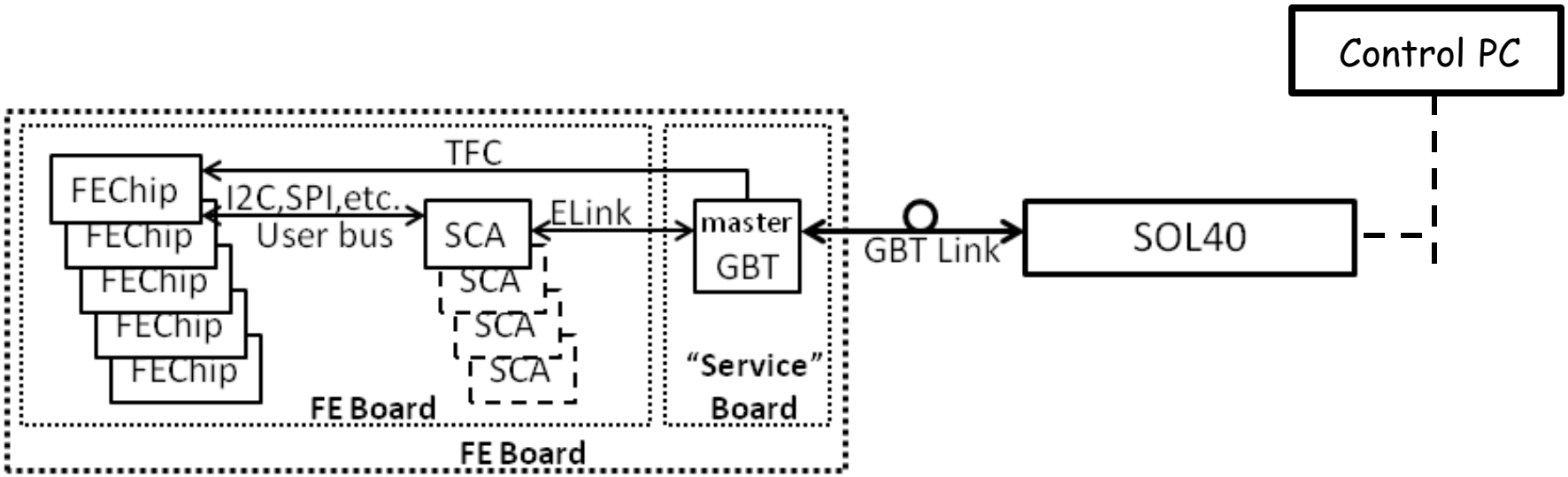


- Two Architectures envisaged:
  - FE electronics in one single FE board
  - FE electronics accessed via a "Service" board (masterGBT <-> SCA via "long" Elink)

FE Board Protocols

- **Interface to the FE Chips**
  - The GBT-SCA provides the following protocols:
    - 16 x I2C master controllers
    - 1 x JTAG master controller
    - 32 x ADC channels (multiplexed)
    - 1 x Memory bus (32 bits) controller
    - 4 x PIA (Parallel Interface Adapter) controllers
    - 1 x SPI (Serial Peripheral Interface) bus
    - 4 x DAC channels
  - **Recommended protocols (for bulk transfers) are:**
    - I2C
    - SPI using independent chip selects (daisy chained SPI not allowed)

FE Addressing

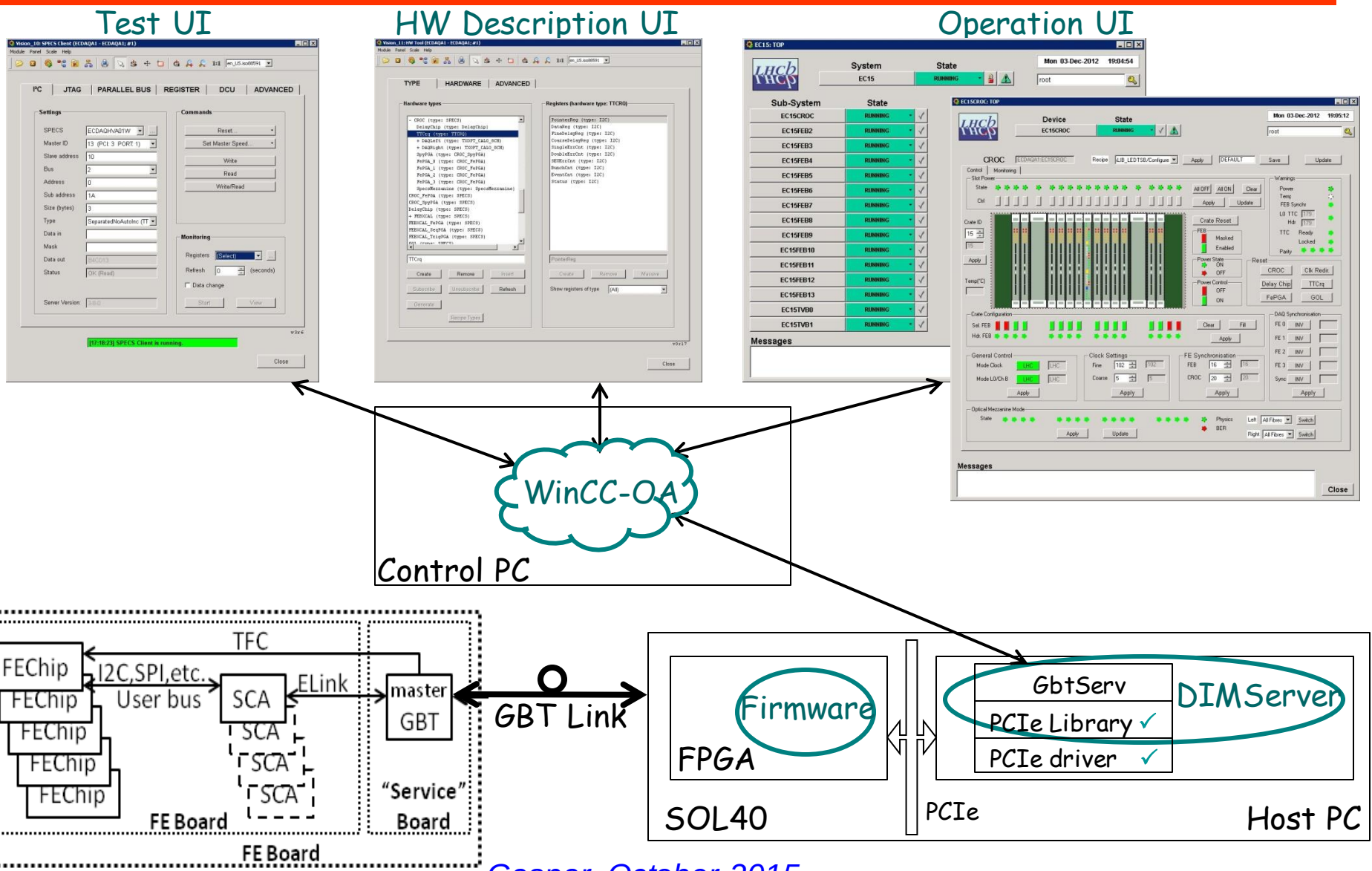


- **FE Chip Register Address:**

<SOL-ip><GBT-i><SCA-j><ProtoCode><I2C-k><I2C-add>[<I2C-s.add>]

fixed **variable**

ECS FE Dataflow

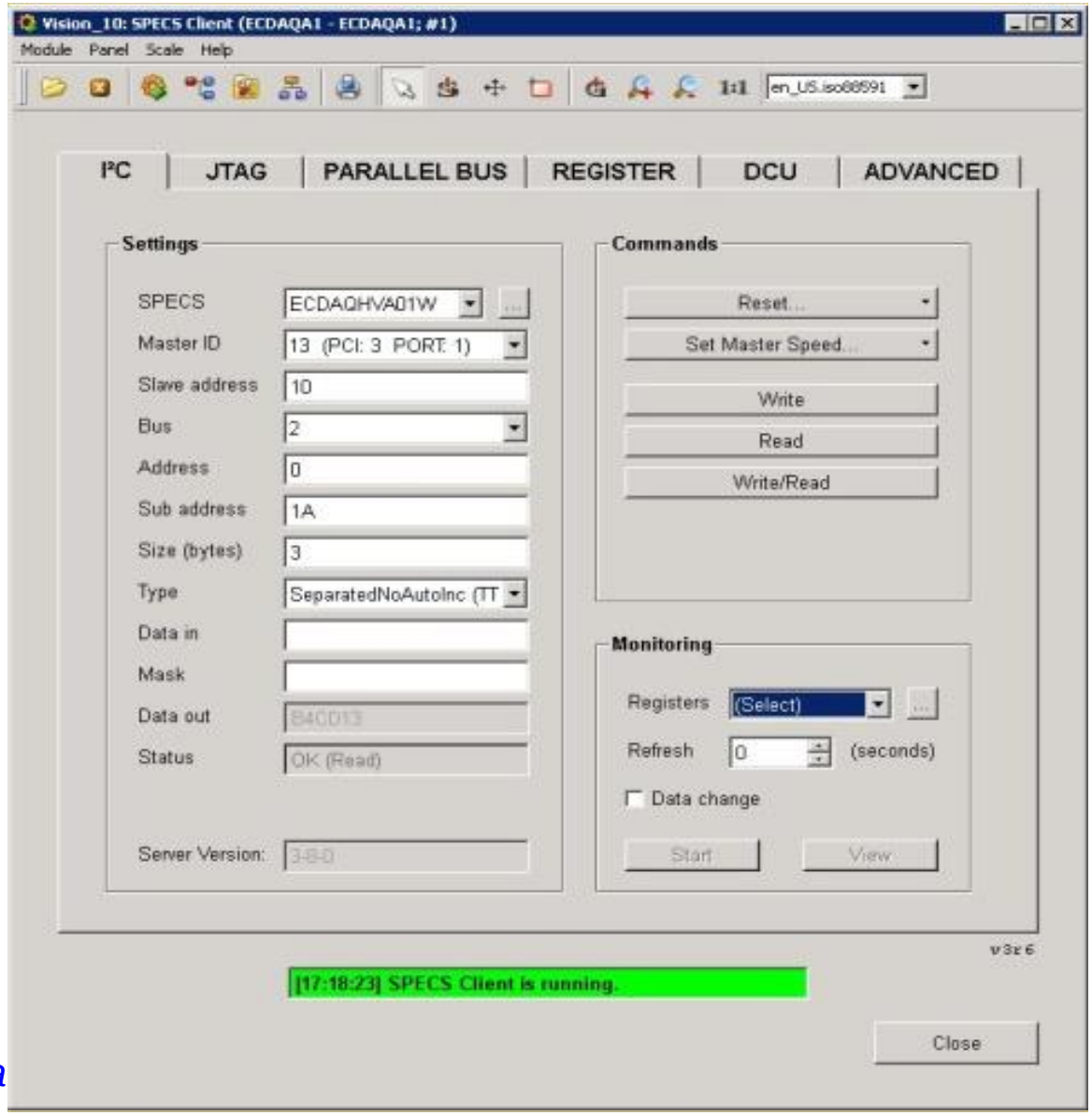


FE ECS Software/firmware

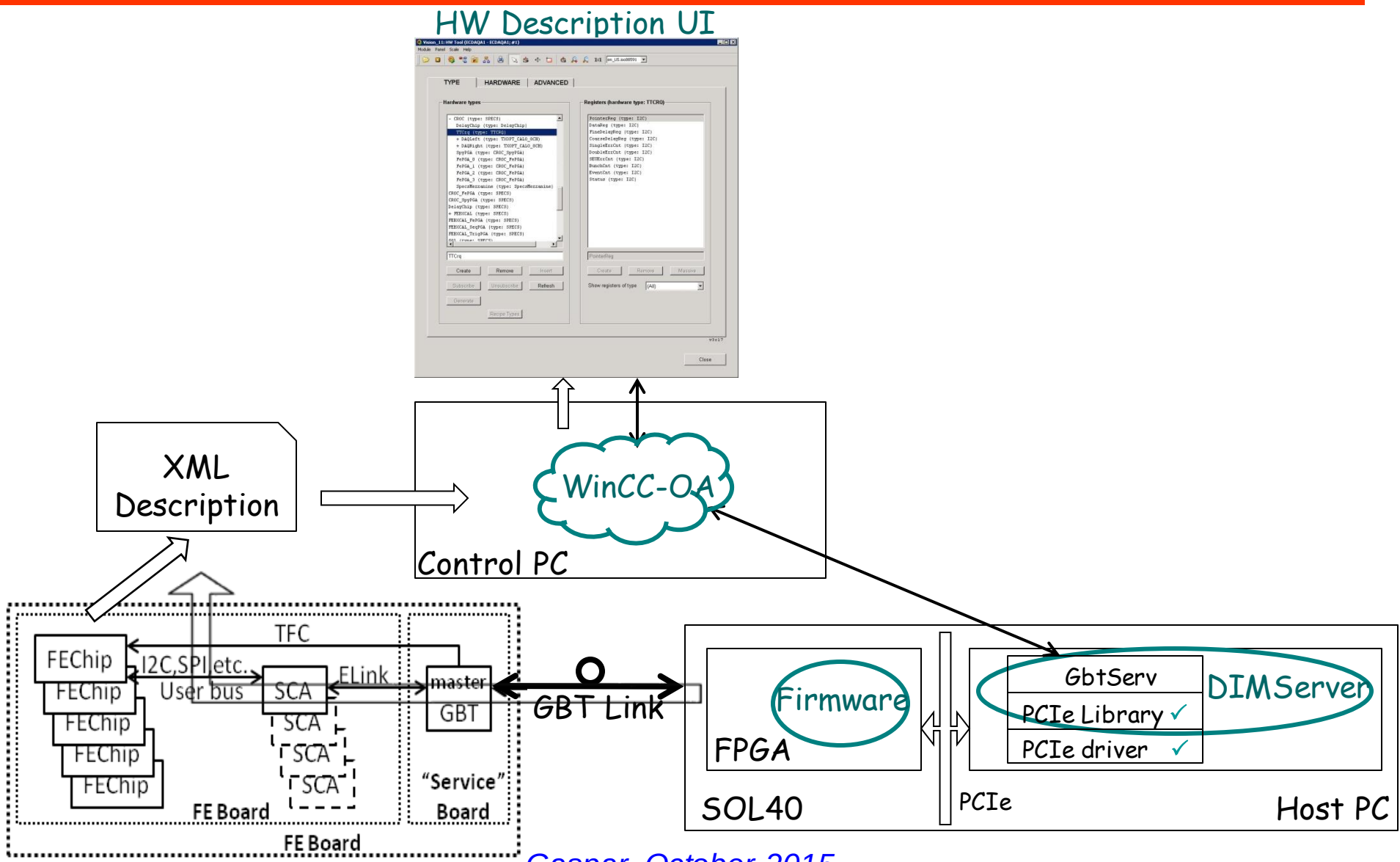
- **Will be centrally provided:**
  - The FPGA firmware for the SOL40 board
    - Will prepare, send and receive the GBT-SCA frames for the various user protocols
      - Should take load away from CPU as much as possible
  - Low-level libraries and command-line tools:
    - Will allow accessing the different FE chips
  - A DIM server:
    - Will implement higher-level commands to configure and monitor the FE chips
  - A WinCC-OA component
    - Providing the high-level description and access of all electronics components

WinCC-OA: Test UI

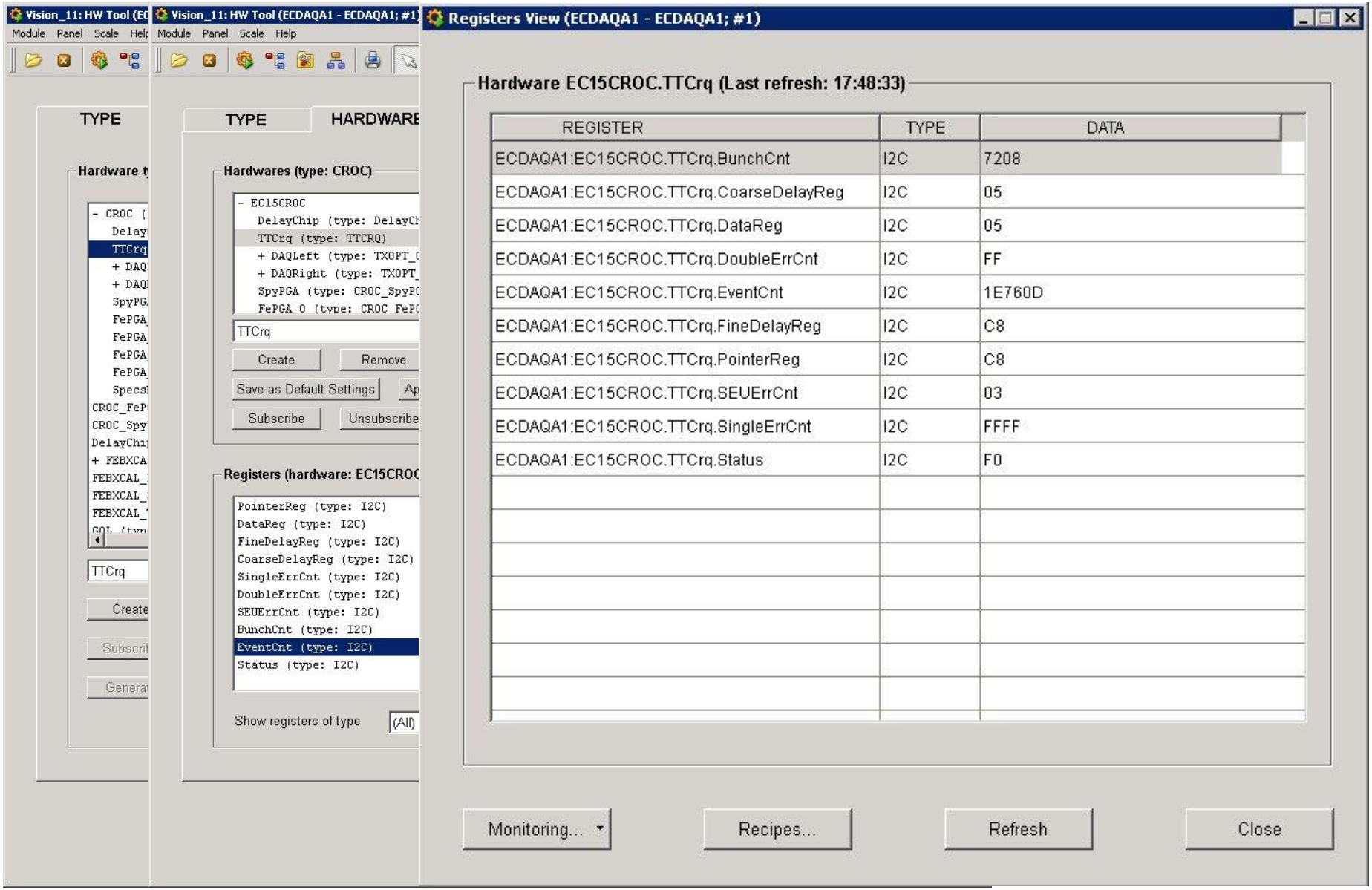
- For both BE and FE boards



WinCC-OA: HW tool

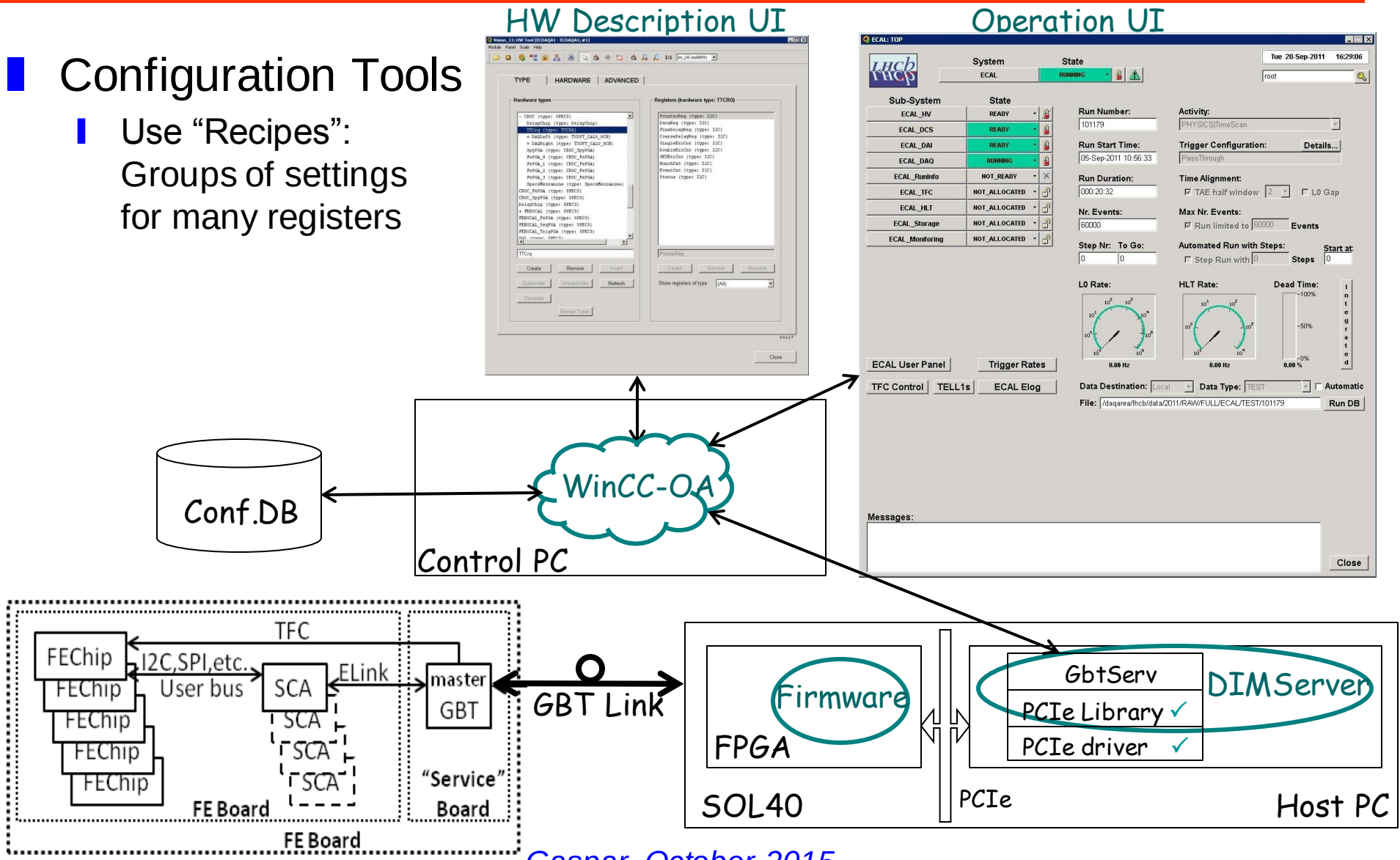


WinCC-OA: HW tool



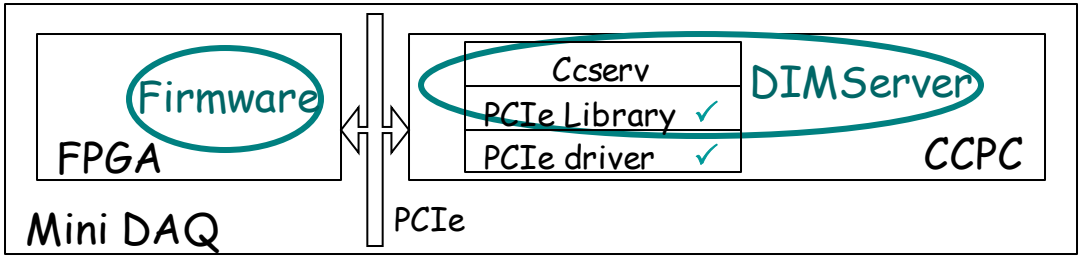
WinCC-OA: Configuration

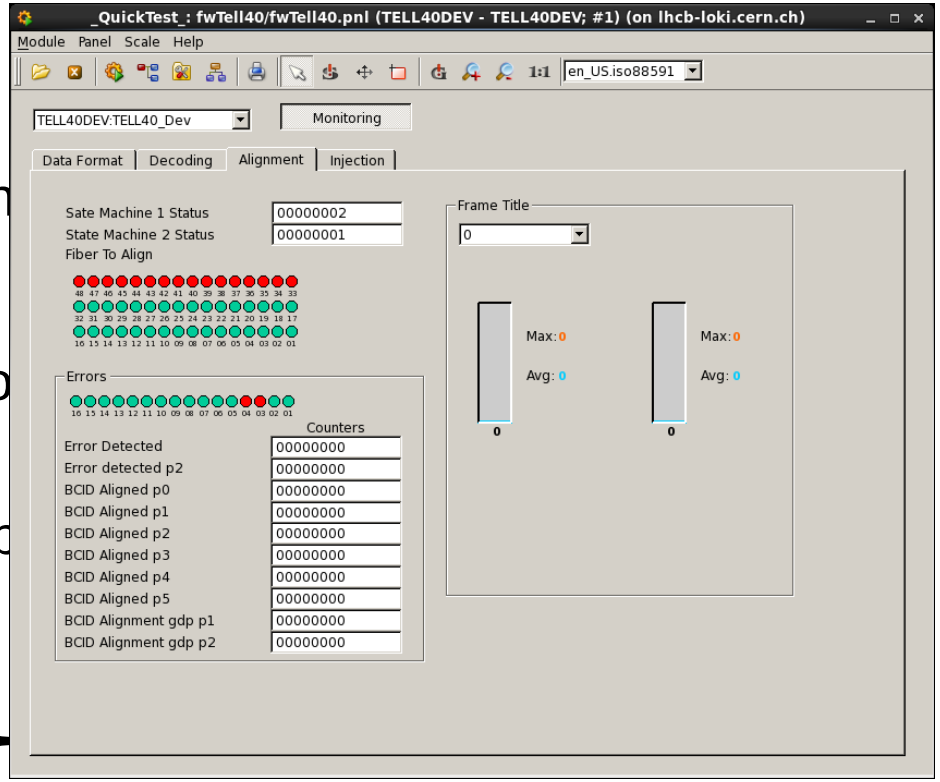
- Configuration Tools
  - Use “Recipes”: Groups of settings for many registers



Current Status

- **MiniDAQ Based**
  - BE boards:
    - DIM Server adapted
    - FwCcpc/FwHw adapted accepts XML input
    - Started building Tell40 pan
  - FE Boards:
    - SOL40 Firmware 1st versio
    - Building low-level libraries
    - Extending the XML descrip
  - In general:
    - FwCcpc / Ccserv (tell1)
    - FwSpecs / SpecsServer





Scalability & Efficiency

- **Configuration and Monitoring speed**
  - Is not just data size / bandwidth
  - Depends on:
    - How the data is distributed, for ex.:
      - A few large registers or many small ones
    - Are blocking operations needed, for ex.:
      - A register needs to be set and read-back before configuring the next one
      - or read-modify-write operations that need to be done by the back-end
    - Which user protocol is used and how it is used, for ex.:
      - Some I2C devices needed an extra I2C operation to write a sub-address before the block of data could be written
      - Some did not implement auto-increment so block transfer could not be used
  - Try to influence design early enough…